

Graph Cut Based Multiple View Segmentation for 3D Reconstruction

Mario Sormann, Christopher Zach, Konrad Karner
VRVis Research Center
Inffeldgasse 16/II
A-8010 Graz, Austria
sormann@vrvis.at

Abstract

In this paper we propose a novel framework for efficiently extracting foreground objects in so called short-baseline image sequences. We apply the obtained segmentation to improve subsequent 3D reconstruction results. Essentially, our framework combines a graph cut based optimization algorithm with an intuitive user interface. At first a meanshift segmentation algorithm partitions each image of the sequence into a certain number of regions. Additionally we provide an intelligent graphical user interface for easy specification of foreground as well as background regions across all images of the sequence. Within the graph cut optimization algorithm we define new energy terms to increase the robustness and to keep the segmentation of the foreground object coherent across all images of the sequence. Finally, a refined graph cut segmentation and several adjustment operations allow an accurate and effective foreground extraction. The obtained results are demonstrated on several real world data sets.

1. Introduction

Robust and accurate foreground/background detection in image sequences is a crucial step in many applications for computer vision and 3D reconstruction [20]. A detailed 3D reconstruction of a complex object requires many images, which in turn demands to segment those images, which can be a tedious and time consuming process.

However, an automatic segmentation procedure is still a challenging problem because of the ambiguity between foreground and background regions. Due to this fact human assistance in combination with a well designed graphical user interface can be essential to obtain a robust and accurate foreground segmentation. Basically, this paper addresses the problem of extracting the foreground object in an image sequence, which is embedded in an arbitrary environment. The main contribution of this work, in contrast

to video segmentation [18], is to deal with a highly varying background as well as foreground as typically provided in image sequences used for 3D reconstruction. Thus, we propose a feasible multiple view segmentation framework based on graph cut optimization. Its power can be derived from the fact that labeled image sequences simplify the correspondence problem dramatically and therefore, dense 3D reconstruction results of complex objects can be substantially improved.

The structure of our system can be roughly separated into the following major tasks. At first, an automatic pre-processing step is initiated to segment each image of the sequence in a certain number of regions by applying mean shift analysis. Since we apply the graph cut optimization on these generated regions, we are able to process large image sequences in acceptable time. Subsequently, a graph cut based optimization framework is combined with a novel graphical user interface, which provides a robust and accurate extraction of a foreground object, whose background can not be removed in a simple way. Moreover, experiments demonstrate that our method produces proper segmentation results, while requiring only little effort from the user.

2. Related Work

Current state of the art foreground object extraction methods can be roughly divided into two main categories: boundary-based segmentation methods and region-based segmentation methods. In the following we focus on those methods which are closely related to our multiple view segmentation framework.

2.1. Boundary-Based Segmentation

In general, boundary-based methods are highly assisted by a human operator. These methods allow a user to trace the foreground object roughly with any input device, e.g. a mouse, whereas the system optimizes the initial object boundary. One of the most popular approaches in this field

are intelligent scissors introduced by Mortenson and Barrett [12] and *JetStream* proposed by Pérez et. al. [13]. The main limitation of these method is the huge amount of necessary user input, especially for highly textured regions, to obtain a satisfying result.

2.2. Region-Based Segmentation

In region-based methods the human operator selects a region inside the foreground object, allowing the system to give the user a foreground suggestion, which can be further modified by selecting additional regions. The simplest region-based segmentation approach is the magic wand, which is part of Adobe Photoshop [14] for many years. This methods utilizes a user-specified point or region inside the object, whereas the system computes a region set of connected pixels that have similar color statistics to the selected one.

Recently developed approaches utilize more sophisticated probabilistic color models and optimization techniques to improve the segmentation results. Well known in this field are graph cut methods, originally invented by Boykov and Jolly [1] and all its derivatives [11, 15]. These methods are based on pixel labeling by energy minimization. The user selects some foreground and background pixels, the so called seed points, with an input device and the system determines the optimal boundary using a minimum-cut/maximum-flow algorithm. The obtained segmentation can be further enhanced by selecting more seed points.

More recently, graph cut algorithms also have been applied to video sequences, introduced by Li et. al. [10] and Wang et. al. [18]. In contrast to our approach they assume a static background, a slightly varying foreground and only minor changes between successive frames, typically provided by video sequences. However, this work is comparable with an earlier approach of foreground extraction to improve a subsequent 3D reconstruction as proposed by Sormann et. al. [17], but utilizes a combination of the mean shift operator with the well known intelligent scissors technique.

In our system, mean-shift segmentation is applied to each image of the sequence to achieve interactive feedback for a human operator. Mean-shift analysis was originally invented by Fukunaga and Hostetler [8] and recently successfully applied to image segmentation and tracking by Comaniciu and Meer [4]. The meanshift analysis approach is defined as a gradient ascent search for maxima in a density function, defined over a high dimensional feature space. The feature space includes a combination of the spatial coordinates and all its associated attributes that are considered during the analysis.

Shape priors combined with graph cuts, as proposed by Freedman and Zhang [6], can further improve segmentation

results. Originally, shape priors are often combined with level set methods [3], which represents a complete different technique for image segmentation. Level set methods are based on front propagation by solving corresponding partial differential equations. Cremers et. al. [5] proposed a variational approach, in which, besides a level set function for segmentation, a new function called labeling function is introduced to indicate the regions in which shape priors should be enforced. We borrow the idea of shape priors to increase the robustness of our multiple image segmentation system.

3. Multiple View Segmentation

The framework of our system is shown in Figure 1. Basically our system proceeds along three major steps, namely an automatic preprocessing step, a coarse graph cut based segmentation task and finally a refined graph cut based segmentation procedure. Additionally, we assume that the input images are taken with short baseline, resulting in smooth transitions of the foreground object between each pair of successive images.

In the preprocessing step the image sequence is divided into a certain number of regions by applying a mean shift segmentation algorithm. Due to the fact that the graph is build on these pre-computed regions, instead of image pixels we improve the efficiency and ensure instant feedback during the segmentation procedure. The second task is assisted by a human operator utilizing our novel user interface, where a user can easily indicate foreground as well as background regions, simultaneously in all images of the sequence. Afterwards, an coarse graph cut based segmentation algorithm is performed considering previous computed foreground/background color signatures and gradient information between neighboring regions. Additionally, to increase the robustness, we embed a shape prior in the optimization framework. Since mean shift segmentation provides an acceptable set of object boundaries, we achieve an appropriate segmentation result where the foreground object is extracted almost correct. Nevertheless, in the final stage a refined pixel-based graph cut segmentation is performed in a small band around the coarse boundary. For the remaining misclassified regions, for example if the foreground/background color difference is negligible small, the user is still in control and able to override the obtained segmentation result.

3.1. Automatic Preprocessing

Since we process large image sequences for dense 3D reconstruction, a pixel-based segmentation, as proposed by Boykov and Jolly [1] is computational impossible. Therefore we pre-segment each image of the sequence with the

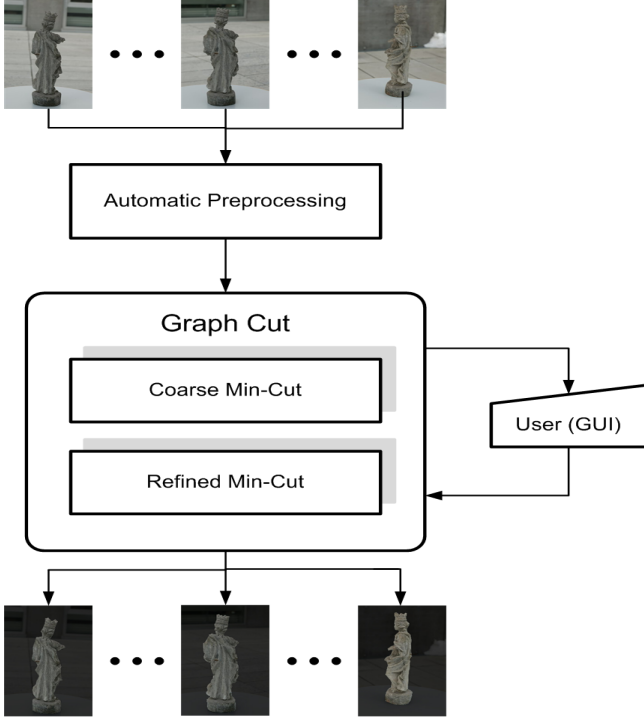


Figure 1. Overview of our multiple view segmentation pipeline: First, the input image sequence is over-segmented during the automatic preprocessing stage. Then, based on human assistance, a coarse graph cut optimization procedure is initiated. The final task consists of an automatic refinement of the foreground object boundary.

meanshift segmentation algorithm and generate the graph for the obtained regions. As shown in Li et.al. [11] such a pre-processing clearly improves the interactive min-cut optimization and instant visual feedback is achievable. Moreover, we build a list of all pairs of adjacent mean shift regions and pre-compute the link costs between each pair of neighboring regions. Further details on link costs will be discussed in Section 3.3.

3.2. User Interface Design

The basic idea of our user interface relies on the fact that the image sequence is considered as a 3D image cube, which is similar to a 3D video cube as proposed by Klein et. al. [9]. They visualized a video in form of a 3D volume. Our user interface design is inspired by this approach, which allows an instant and simultaneous access to all images of the sequence. Consequently, we provide three views to realize a simple and fast indication of foreground/background

regions within the image sequence.

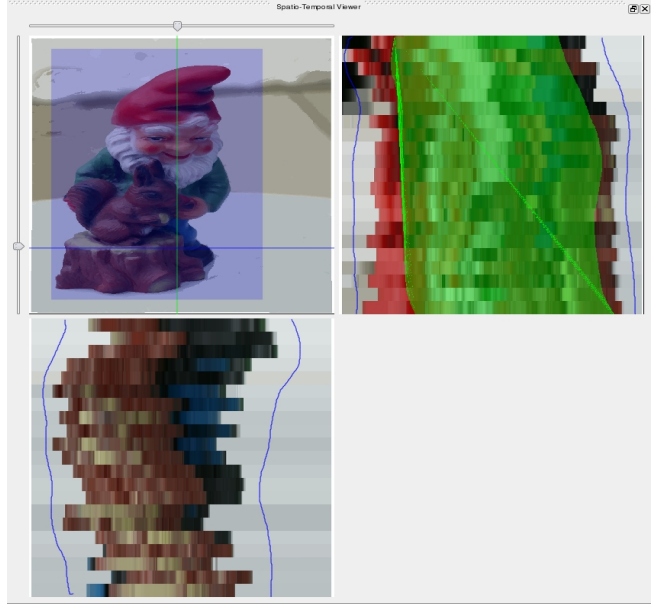


Figure 2. Illustration of our implemented user interface, which splits the image stack into three views, namely a xy-viewer (top left), a xs-viewer (bottom-left) and a ys-viewer (top right). The xy-viewer is represented by a standard image viewer, whereas the xs/ys-viewer represents a cut through the image stack at a defined position. The blue curves indicate the user selected region of interest, which can be verified in the xy-viewer (shown as blue overlay). The marked foreground regions are highlighted as a green polygon in the ys-viewer.

The image sequence is visualized as a 3D texture, where the horizontal and vertical dimensions of the images are covered by the X- respectively Y-axis, whereas the Z-axis represents the stack of input images. Figure 2 shows our implemented user interface, which contains three types of windows: xy-viewer (top left), xs-viewer (bottom left) and a ys-viewer (top right). In contrast to the xy-viewer, which represents a standard image viewer, the xs/ys-viewer covers a cut at a defined position through the image stack. The cut position can be interactively changed via sliders of the xy-viewer, which allow a cut at any position through the image stack. This enables a simultaneous access to all input images for a human operator.

The required input data to initiate the coarse graph cut optimization are confident background regions, based on a region of interest covering the desired object, as well as a certain number of reliable foreground regions. For the re-

gion of interest the user simply draws curves (blue curves) over the x_s -viewer, respectively y_s -viewer, which enclose the desired foreground object. Each curve in the x_s -viewer and y_s -viewer defines a set of points in the corresponding xy -viewer to build our required region of interest for the complete image stack. Since the user can move through the image stack, the correctness of the obtained background regions can be easily verified at any time during the labeling procedure, shown as blue overlay in the xy -viewer in Figure 2. On the other hand, the foreground regions are extracted directly from the input viewers, exemplary illustrated as green polygon in the y_s -viewer of our user interface.

After all, the user initiates the coarse graph cut optimization, which extracts the foreground object in all images with reasonable speed. Naturally, if the segmentation result is not satisfying the user can add more foreground regions and perform graph cut optimization again.

3.3. Coarse Graph Cut Optimization

The multiple view segmentation problem is treated as a binary labeling problem of the input image stack. Basically, each image is represented by a graph $G = \langle V, E \rangle$, where V is the set of all nodes represented by the mean shift regions and E is the set of all adjacent nodes, respectively all adjacent mean shift regions. As known, the labeling problem is to assign each region a unique label $r \in \{0(background), 1(foreground)\}$. In general, the solution can be obtained by energy minimization formulated as a graph cut problem, discussed by Boykov and Kolmogorov [2] in more detail. For our graph cut optimization the labeling problem is solved by minimizing the following energy function $E(X)$:

$$E(X) = \sum_{r \in V} E_i(x_r) + \lambda_1 \sum_{(r,s) \in E} E_i(x_r, x_s) + \lambda_2 \sum_{(r,s) \in E} E_s(x_r, x_s) \quad (1)$$

where $E_i(x_r)$ encodes the color similarity of region r to the foreground/background color signature generated from previous selected foreground/background regions. The second energy term $E_i(x_r, x_s)$ measures the color gradient between adjacent nodes r and s . The third term $E_s(x_r, x_s)$ incorporates a shape prior obtained from previous image in the graph cut optimization process. The weighting terms are set in practice to $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$. An outline of our developed graph cut nodes is illustrated in Figure 3.

3.3.1 Likelihood Energy $E_i(x_r)$

The likelihood energy encodes the similarity of the color of a node x_i to the selected foreground seed regions and

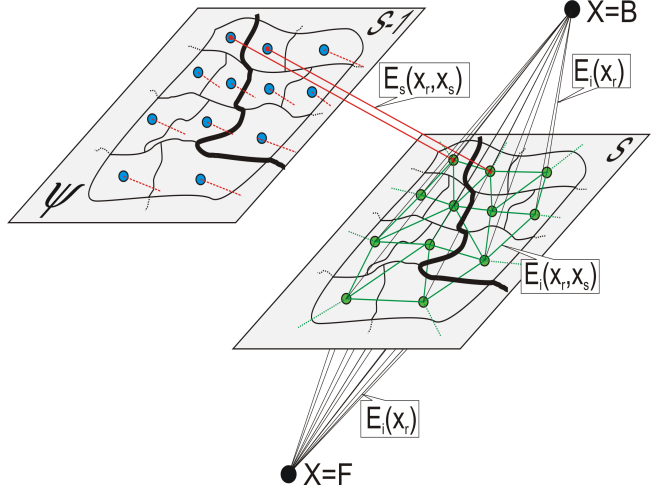


Figure 3. Outline of our implemented graph cut nodes and their internal relationships. According to the energy term $E_i(x_r)$ all regions are connected to the virtual background ($x = B$) and foreground ($x = F$) node called source and sink respectively. Furthermore $E_i(x_r, x_s)$ link neighboring regions (green edges), whereas $E_s(x_r, x_s)$ incorporate the shape prior energy obtained from the previous segmented image to the current regions (red edges), which enforces temporal coherence.

background seed regions, denoted F and B respectively. So far the user had already indicated these two sets of regions in the image sequence, as discussed in Section 3.2. All other regions are marked as uncertain regions, labeled U . The basic idea behind our similarity estimation is to build a color signature of the known background as well as of the known foreground, similar to the one described in Friedland et.al. [7]. Essentially, the known regions are clustered in the CIELAB space with a two stage k-d tree algorithm, as proposed by Rubner et. al. [16]. The CIELAB color space is explicitly designed to approximate human color discrimination. In the first stage, initial clusters are found by subdivision and stops when a cell becomes smaller than a predefined threshold. Due to the fact that a cluster may be distributed over a few cells the second stage recombines them. Finally, the obtained color signatures for foreground S_j^F and background S_k^B are used to compute the likelihood energy $E_i(x_r)$ as follows:

	$r \in F$	$r \in B$	$r \in U$
$E_i(x_r = F)$	0	∞	$\frac{d_r^F}{d_r^F + d_r^B}$
$E_i(x_r = B)$	∞	0	$\frac{d_r^B}{d_r^F + d_r^B}$

Assignments of 0 and ∞ enforce consistent segmentation according to user input, whereas the distance $d_r^F = \min\|C(r) - S_j^F\|$ encodes the minimum distance of the color of region $C(r)$ to the foreground color signature, respectively $d_r^B = \min\|C(r) - S_k^B\|$ is defined accordingly.

3.3.2 Prior Energy $E_i(x_r, x_s)$

The prior energy $E_i(x_r, x_s)$ measures the gradient along the foreground object. To increase the robustness, we extend the energy term, as proposed in former approaches as follows:

$$E_i(x_r, x_s) = \frac{1}{1 + \kappa} \quad (2)$$

$$\kappa = \lambda_3 \cdot C_{rs} + \lambda_4 \cdot \nabla_{rs} \quad (3)$$

Here C_{rs} is the $L2$ -Norm of the color difference of two regions r and s and ∇_{rs} represents the local gradient intensity on the shared boundary between region r and s . Note, that the prior energy is non-zero only when the selected nodes are set to contrary labels, e.g. $x_r = B$ and $x_s = F$. In all our experiments the weighting factors are fixed to $\lambda_3 = 0.6$ and $\lambda_4 = 0.4$.

3.3.3 Shape Prior Energy $E_s(x_r, x_s)$

Freedman and Zhang [6] already introduced a segmentation framework for still images, which incorporates shape priors in a graph cut based algorithm. We adopt their idea and extend it to the more challenging problem of multiple view segmentation. Consequently, in our framework the shape prior energy is designed to keep the segmentation of the foreground object coherent across all images of the sequence. We embed the shape knowledge as shape prior energy to the graph cut optimization process, which clearly increases the robustness of our multiple view segmentation framework.

Basically, we specify the shape prior as a regular unsigned distance function and propagate as reference shape the extracted contour of the previous image to the current segmentation procedure. Due to the fact that the coarse graph cut optimization is performed on image regions we further convert the pixel-based distance function to a region-based distance function, based on simple averaging of the distance values on the region boundary. In case of varying positions of the foreground object in the image sequence, a translation of the template, utilizing our current region of interest, might be necessary.

Obviously, for the first image of the sequence no reference shape can be extracted and the shape prior energy is set to zero. Now, the shape prior energy is defined as follows:

$$E_s(x_r, x_s) = \sum_{r,s \in E} \Psi_{trans}(\frac{r+s}{2}) \quad (4)$$

where Ψ_{trans} tends to be small near the shape template and thus incorporates shape coherence between adjacent image pairs in the graph cut optimization process.

As mentioned, the objective function of Equation 1 is finally minimized using the min-cut algorithm in [2]. Figure 4 illustrates all implemented energy terms, visualized on an image of the equestrian statue.

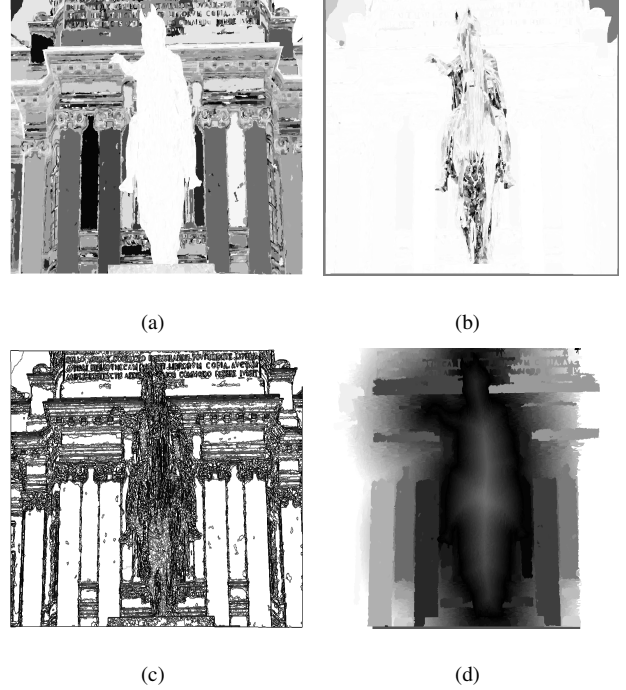


Figure 4. Illustration of implemented energy terms, visualized on one image of the equestrian statue data set. (a) Foreground likelihood energy $E_i(x_r = F)$. (b) Background likelihood energy $E_i(x_r = B)$. (c) Prior energy $E_i(x_r, x_s)$ (d) Shape prior energy $E_s(x_r, x_s)$. Note, that all energies are globally minimized, utilizing graph cut optimization, which allows us to separate the foreground object in a complex moving background.

3.4. Refined Graph Cut Optimization

Since the coarse graph cut optimization is based on mean shift regions, the boundary of the extracted foreground object is usually somehow jagged. Additionally, there can be still some missegmentation, especially around low contrast edge boundaries. The refined graph cut optimization is designed to refine the obtained coarse segmentation and further provide simple tools to correct produced missegmentation.

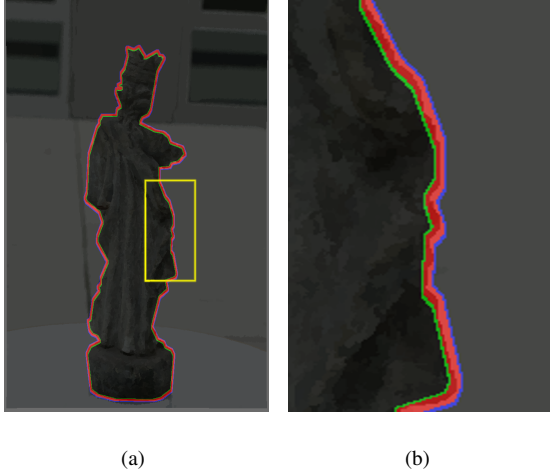


Figure 5. Illustration of refined graph cut segmentation in a small band around the initial segmentation, showing a statue of St. Barbara. (a) Only pixels in the red band are considered for the pixel based graph cut optimization. (b) Close-up of (a) showing foreground seeds (green) and background seeds (blue).

Unlike the first stage, we perform a pixel based graph cut within a narrow band of the coarse boundary to refine the border. First of all, the coarse boundary is simplified into an editable polygon. Then, we build in a small band around the simplified polygon a pixel-based graph:

$$E(X) = \sum_{p \in V_B} E(x_p) + \lambda_1 \sum_{(p,q) \in E_B} E(x_p, x_q) \quad (5)$$

where V_B are all pixels in the small band and E_B represents the eight-neighborhood between pixels. In our experiments the width of the band is fixed to eight pixels. Likelihood energy $E(x_p)$ and prior $E(x_p, x_q)$ are defined as in the coarse graph cut optimization, except that the regions are replaced by pixels. The seed pixels to compute the foreground as well as background color signature are acquired from the inner boundary band, respectively from the outer boundary band, as illustrated in Figure 5. λ_1 is identical to the weighting term in Equation 1.

4. Experiments

All presented experiments are performed on a 3.4 GHz PC with 2GB main memory and a GeForce 7800 GT with 256MB graphics memory. The image sequences are taken with a calibrated digital consumer camera. The image res-

olution of the demonstrated data sets is fixed to 2032x1352 pixels.

Table 1 illustrates the complexity and processing performance of four real world image sequences. The automatic preprocessing time was approximately 35-45% of the overall segmentation time, whereas the time for coarse and refined graph cut optimization is acceptably small. Human operator time correlates directly with the complexity of the scene, thus complex background or low contrast edges typically yield to more user interaction.

Figure 6(a) shows the garden gnome data set. The garden gnome is approximately 23cm tall with a diameter of 10cm. As the image sequence consists of high contrast edges and large color differences between foreground and background, almost no human assistance was necessary to achieve a reasonable segmentation result. More challenging data sets are illustrated in Figure 6(b-d). In the case of the statue of St.Barbara our framework has to deal with quite similar foreground and background colors, thus more human assisted overriding operations are performed. This statue is 55cm tall with a diameter of 13cm at the pedestal. The ceremonial room statue exhibits a very complex background and partially low contrast edges. Similar problems are obtained for the equestrian statue data set. Both objects are approximately 2.5m high.

As already mentioned, sometimes our multiple view segmentation framework fails and produces incorrect segmentation results. Therefore we provide several simple tools, which allow a user to directly control the object boundary. As in other approaches we allow direct vertex editing and overriding operations. If a user prefers boundary based techniques, we incorporate the so called live wire brush, which replaces a specified segment of the objects boundary. Such a tool can be especially useful for thin structures, where a direct boundary control is clearly helpful. The last column of Table 1 reports the used time for these operations.

As soon as the foreground object is extracted an optional GPU-based dense matching [19] is initiated. Moreover, after volumetric range image integration we obtain a complete 3D model of the extracted foreground object. On the right hand side of Figure 6 generated depth maps as well as the complete 3D reconstructions are shown.

5. Conclusion

In this work we present a graph cut based foreground extraction framework, that performs well in terms of efficiency, robustness and accuracy. The primary purpose of our method is the improvement of our 3D reconstruction results. Moreover, the tedious task of an interactive segmentation of all images are clearly reduced, since our novel user interface provide a direct access to all images across the sequence.

Dataset	Images	Pre-Pr. [sec.]	C. Min-Cut [sec.]	R. Min-Cut [sec.]	U. S. [min.]	U. O. [min.]
Garden gnome	21	≈ 195	8.4	13.23	≈ 3	≈ 1
St. Barbara	42	≈ 380	21.1	27.3	≈ 4	≈ 10
Ceremonial statue	19	≈ 175	12.35	12.73	≈ 3	≈ 8
Equestrian statue	55	≈ 500	30.25	34.1	≈ 4	≈ 16

Table 1. Illustration of complexity and processing performance of four real world data sets. In contrast to coarse min-cut (C. Min-Cut) and refined min-cut (R. Min-Cut), automatic preprocessing (Pre-Pr.) takes 35-45% of the overall processing time. The rest of the processing time is assigned to user interaction indicating foreground/background seeds (U. S.) with our previous discussed user interface and overriding operations (U. O.) to correct segmentation errors.

Though the results are very promising, there are several improvements than can be made to our approach. In order to further increase the robustness, incorporation of the relative pose information of the cameras might help to achieve this goal. Another interesting problem for future work is to develop 3D adjustment tools, in order to correct faulty segmentation over several images simultaneously.

Acknowledgments

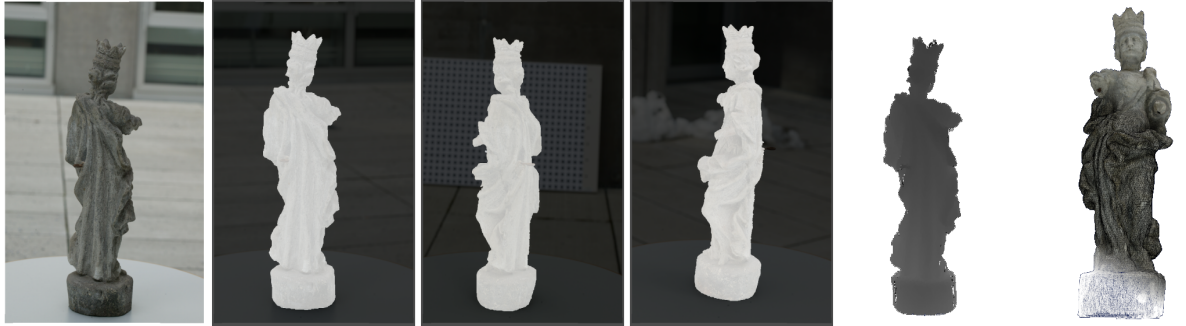
This work is partly funded by the VRVis Research Center, Graz and Vienna/Austria (<http://www.vrvis.at>). We would also like to thank the Vienna Science and Technology Fund (WWTF) for supporting our work in the *Creative Histories - The Josefsplatz Experience* project.

References

- [1] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *International Conference of Computer Vision*, volume 1, pages 105–112, Vancouver, Canada, July 2001.
- [2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137, September 2004.
- [3] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. In *International Conference of Computer Vision*, volume 1, pages 694–699, Boston, USA, June 1995.
- [4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002.
- [5] D. Cremers, T. Kohlberger, and C. Schnoerr. Nonlinear shape statistics in mumford-shah based segmentation. In *European Conference on Computer Vision*, volume 2351, pages 93–108, Copenhagen, Denmark, May 2002.
- [6] D. Freedman and T. Zhan. Interactive graph cut based segmentation with shape priors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 755–762, San Diego, USA, June 2005.
- [7] G. Friedland, K. Jantz, and R. Rojas. Siox: Simple interactive object extraction in still images. In *IEEE International Symposium on Multimedia*, volume 1, pages 253–259, December 2005.
- [8] K. Fukunaga and L. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, January 1975.
- [9] A. W. Klein, P. J. Sloan, A. Finkelstein, and M. F. Cohen. Stylized video cubes. In *ACM SIGGRAPH Symposium on Computer Animation*, volume 1, pages 15–22, July 2002.
- [10] Y. Li, J. Sun, and H.-Y. Shum. Video object cut and paste. *ACM Transactions on Graphics*, 24(3):595–600, July 2005.
- [11] Y. Li, J. Sun, C. Tang, and H.-Y. Shum. Lazy snapping. *ACM Transactions on Graphics*, 23(3):303–308, August 2004.
- [12] E. N. Mortenson and W. Barrett. Intelligent scissors for image composition. *Graphical Models and Image Processing*, 60(5):349–384, September 1998.
- [13] P. Pérez, A. Blake, and M. Gangnet. JetStream: Probabilistic contour extraction with particles. In *International Conference of Computer Vision*, volume 2, pages 524–553, Vancouver, Canada, July 2001.
- [14] M. Perkins. *Beginner's Guide to Adobe Photoshop*. Amherst Media, 2 edition, September 2003.
- [15] C. Rother, V. Kolmogorov, and A. Blake. Grabcut - interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, August 2004.
- [16] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, November 2000.
- [17] M. Sormann, C. Zach, J. Bauer, K. Karner, and H. Bischof. Automatic foreground propagation in image sequences for 3d reconstruction. In *Annual meeting of the German Association for Pattern Recognition*, volume 1, pages 93–99, August 2005.
- [18] J. Wang, P. Bhat, R. Colburn, M. Agrawala, and M. F. Cohen. Interactive video cutout. *ACM Transactions on Graphics*, 24(3):585–594, July 2005.
- [19] C. Zach, M. Sormann, and K. Karner. High-performance multi-view reconstruction. In *International Symposium on 3D Data Processing, Visualization and Transmission*, Chapel Hill, USA, June 2006.
- [20] R. Ziegler, W. Matusik, H. Pfister, and L. McMillan. 3d reconstruction using labeled image regions. In *Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, volume 1, pages 248–259, Granada, Spain, September 2003.



(a)



(b)



(c)



(d)

Figure 6. Original image on the left, multiple view segmentation results in the middle and achieved 3D reconstruction on the right. (a) Garden gnome image sequence. (b) Statue of St. Barbara. (c) Ceremonial room statue. (d) Equestrian statue.